## Plugins模块：

Plugins/SSLStrategy:用来提供策略的模块，

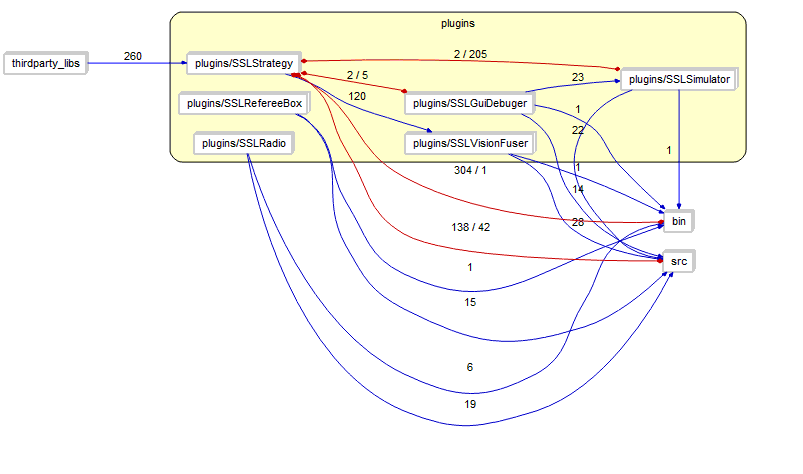
Plugins/SSLVisionFuser:用来对经过局域网收到的文件包加以信息提取，得到场上12辆小车的位置，方向和角速度等信息。

Plugins/SSLRadio:通讯模块，用来收发文件信息

Plugins/SSLGuiDebugger:图像可视化模块

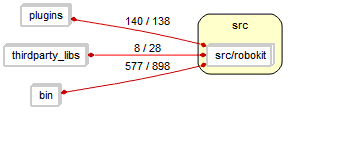
Plugins/SSLRefereBox:裁判盒，用来发出指令

模块静态调用拓扑图如下：

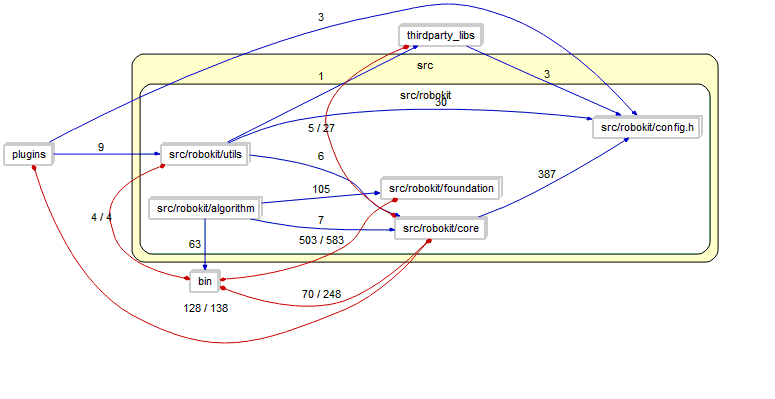


## Src模块

Src:源文件夹，是整个程序的核心控制部分.src文件与plugins, thirdparty\_libs以及bin文件夹均会有相互调用。下面是拓扑图：

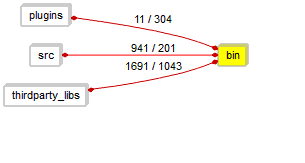


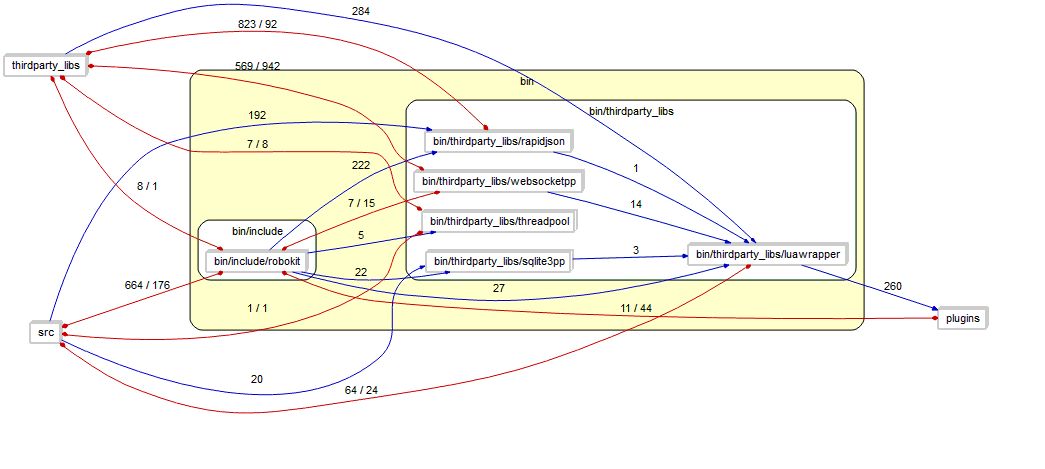
Src文件夹展开如下图所示，有配置文件，有基础组件和核心组件，以及算法部分和工具包组件。调用拓扑关系如下：



## Bin模块

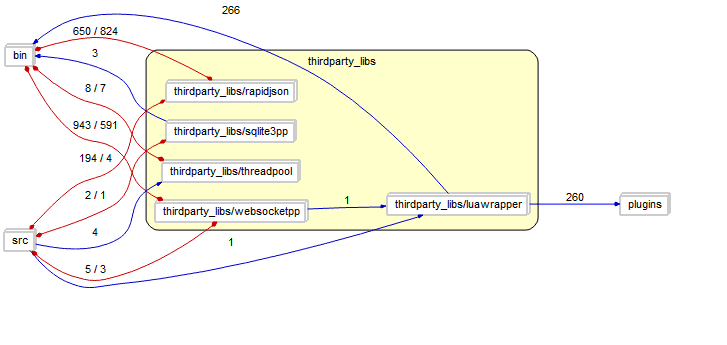
Bin:用来存在二进制文件，包括最后编译生成的exe文件。与src文件夹相反，可以看到bin文件夹与thirdparty\_libs 调用极为频繁，因为程序运行时需要大量的动态链接库，而这些库都被封装在了thirdparty\_libs文件夹中。拓扑图如下所示。





## Thirdparty\_libs模块

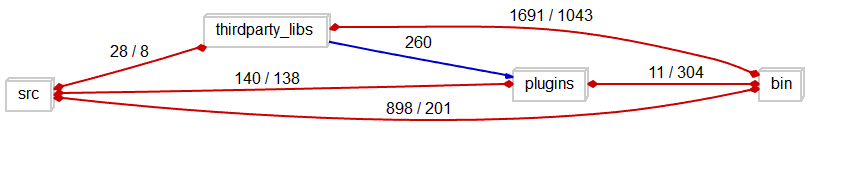
Thirdparty\_libs:第三方库，用来进行底层函数的具体实现，例如websocketpp文件是用来保证网络协议和端口的一致性等与通讯相关的底层部分，threadpool是线程池，用于管理线程，sqlite3pp是与数据库管理相关的子模块。



## 程序框架及运行过程

Lua语言与C++语言的连接实现：lua脚本是一种轻量化脚本语言，用来调用相关函数的接口，底层实现是利用上述的各种C++代码来完成，优势是修改脚本后无需编译即可运行，方便实时快速便捷地调试。

整个程序的框架如下图所示：



Src文件中的源程序编译后会在bin文件夹中生成可执行文件，执行后会与thirdparty\_llibs以及plugins等相互调用，通过plugins实现各种功能，具体的底层实现则是依赖于thirdparty\_libs中的动态链接库。其中的线程池模块控制多线程，一共有三个线程，分别是网络通讯，图像处理，裁判指令以及策略生成。

网络通讯由SSLRadio实现，实现时会调用thirdparty\_libs中的websocketpp；

图像处理分为两种情况，如果是模拟状态，则由SSLVison生成球和车的坐标，如果是实车状态，则由SSLVisionFuser实现；

裁判指令过程由SSLRerefeeBox分析得出相应的指令并传递给主程序；

策略生成则是由SSLStrategy根据赛场上不同的情况并考虑到裁判盒的指令得到不同的策略，用来指导球员的下一步运动。